Technology and Product Reports

An Automated System for Monitoring Soil Moisture and Controlling Irrigation Using Low-cost Open-source Microcontrollers

Rhuanito Soranz Ferrarezi[1], Sue K. Dove, and Marc W. van Iersel

Additional index words. **capacitance sensors, gardening, greenhouse, *Hibiscus acetosella*, nursery, water-saving technologies**

Summary. **Substrate volumetric water content (VWC) is a useful measurement for automated irrigation systems. We have previously developed automated irrigation controllers that use capacitance sensors and dataloggers to supply plants with on-demand irrigation. However, the dataloggers and accompanying software used to build and program those controllers make these systems expensive. Relatively new, low-cost open-source microcontrollers provide an alternative way to build sensor-based irrigation controllers for both agricultural and domestic applications. We designed and built an automated irrigation system using a microcontroller, capacitance soil moisture sensors, and solenoid valves. This system effectively monitored and controlled VWC over a range of irrigation thresholds (0.2, 0.3, 0.4, and 0.5 $m^3 \cdot m^{-3}$) with 'Panama Red' hibiscus (*Hibiscus acetosella*) in a peat:perlite substrate. The microcontroller can be used with both regular 24-V alternating current (AC) solenoid valves and with latching 6- to 18-V direct current (DC) solenoid valves. The technology is relatively inexpensive (microcontroller and accessories cost $107, four capacitance soil moisture sensors cost $440, and four solenoid valves cost $120, totaling $667) and accessible. The irrigation controller required little maintenance over the course of a 41-day trial. The low cost of this irrigation controller makes it useful in many horticultural settings, including both research and production.**

Greenhouse, nursery and floriculture operations collectively generate sales over $16.6 billion and employ 4.9% of hired farm workers in the United States [U.S. Department of Agriculture (USDA), 2010]. In greenhouses, the greatest operating cost is labor (USDA, 2010), and technological innovations, which reduce labor dependency, can help growers stay competitive. Automation can be achieved using technologies such as robots, sensors, and computer-controlled systems, which may also improve plant growth and aesthetic attributes and help growers meet environmental regulations (Majsztrik et al., 2011).

Greenhouse operations traditionally use large quantities of irrigation water and fertilizer to maximize plant growth, product quality, and profits (Conover and Poole, 1992). Irrigation frequency is typically based on a predetermined schedule or visual evaluation of plant water status (Ferrarezi et al., 2014), and this may lead to overwatering, which causes leaching and runoff. The release of water and nutrients from greenhouses results in surface and groundwater contamination and may contribute to eutrophication and environmental degradation (Majsztrik et al., 2011). Sensor-based irrigation can be used to minimize or eliminate leaching from containers by precisely regulating the amount of water used for irrigation (Crespo and van Iersel, 2011; Nemali and van Iersel, 2006).

Sensor-controlled irrigation can be based on a variety of measurements, including evaporation (Allen et al., 1998), substrate VWC (Nemali and van Iersel, 2006), and water tension (Shock and Wang, 2011). In these systems, plant water requirements are estimated based on the measured parameters and irrigation is provided accordingly (Blonquist et al., 2005). Because capacitance soil moisture sensors require less maintenance and provide data that are easier to interpret than tensiometers, VWC may be the most useful measurement for automating irrigation systems (Nemali et al., 2007). Capacitance sensors allow for easy automation (Jones, 2004) and such automated irrigation is reliable (Nemali and van Iersel, 2006) and can be used with a variety of container sizes (Shock and Wang, 2011) and irrigation systems.

A basic system can be made with capacitance sensors, a microcontroller,

Units

To convert U.S. to SI, multiply by	U.S. unit	SI unit	To convert SI to U.S., multiply by
29.5735	fl oz	mL	0.0338
0.0283	ft^3	m^3	35.3147
3.7854	gal	L	0.2642
2.54	inch(es)	cm	0.3937
25.4	inch(es)	mm	0.0394
6.8948	psi	kPa	0.1450
(°F – 32) ÷ 1.8	°F	°C	(°C × 1.8) + 32

Department of Horticulture, University of Georgia, 1111 Miller Plant Science Building, Athens, GA 30602

Funding for this research was provided by USDA-NIFA-SCRI (award no. 2009-51181-05768).

We thank Prof. Dr. Roberto Testezlaf and Shuyang Zhen for critical reviews of this manuscript and Capes Foundation (Ministry of Education, Brazil) for a post-doctoral scholarship to the first author at the University of Georgia (BEX 2620/13-8).

[1] Corresponding author. E-mail: rhuanito@terra.com.br.

and solenoid valves, which open when VWC measurements drop below a predetermined threshold value. Such systems have been successfully used for small-scale experiments with several greenhouse crops, including gaura [*Gaura lindheimeri* (Burnett and van Iersel, 2008)], petunia [*Petunia* ×*hybrida* (van Iersel et al., 2010)], and 'Panama Red' hibiscus (Bayer et al., 2013). More advanced systems have recently been tested in production nurseries (Chappell et al., 2013; Lea-Cox et al., 2013). Cayanan et al. (2008) describe the design of a wireless, automated irrigation system that integrates with the greenhouse environmental control system. Although precision irrigation systems can drastically improve water and nutrient management in greenhouses and container nurseries, they often rely on high-precision dataloggers and computers, making those systems expensive. In addition, they typically require expertise to install and maintain (Crespo and van Iersel, 2011). Our goal was to develop a low-cost, stand-alone irrigation controller.

Low-cost open-source microcontrollers can be used to automate sensor-based irrigation. Simple and reliable microcontrollers with broad functionality have been available since 2005, when Arduino (Ivrea, Italy) made the first of its models available. Silva et al. (2013) describe the use of a microcontroller (Fio, Arduino) to build an irrigation controller, using a cheap soil moisture sensor (less than $5). However, the soil moisture sensor they used (SEN0114; DFRobot, Shanghai, China) is low tech, has exposed electronics, and does not appear to be waterproof. This sensor measures electrical conductivity as a proxy for VWC, which makes the sensors extremely sensitive to salts in the substrate or soil. Silva et al. (2013) also did not test how well their controller could maintain various VWC levels. Recently, Bitella et al. (2014) described the use of a low-cost open-source hardware platform, using another microcontroller (Mega 2560 R3, Arduino), for monitoring soil water content and multiple soil, air, and canopy parameters. However, they did not control irrigation based on the collected data; it was used for monitoring purposes only.

Our objective was to use open-source microcontrollers with capacitance soil moisture sensors to monitor and log VWC and control irrigation based on real-time measurements. This low-cost technology can easily be installed and used in a variety of settings including greenhouse and nursery production or research, home gardens, lawns, and small farms.

Hardware

We constructed a fully automated sensor-based irrigation system controlled by a microcontroller (Uno R3, Arduino) (Table 1, Fig. 1A) with a secure digital (SD) shield and built-in real-time clock [RTC (model 2.0; Adafruit, New York, NY)] (Table 1, Fig. 1C). The microcontroller was connected to a light-emitting diode (LED), a 5-V DC relay driver board with eight relays (SainSmart, Leawood, KS) (Table 1, Fig. 1D), four capacitance soil moisture sensors (10HS; Decagon Devices, Pullman, WA) (Table 1, Fig. 1E), one 6- to 18-V DC 1-inch latching solenoid valve (GCS 3050/3051; Galcon, San Rafael, CA), and three regular 24-V AC 1-inch solenoid valves (Rain Bird, Tucson, AZ) (Table 1, Fig. 1F).

The microcontroller was used to build an irrigation system that can control valves based on capacitance sensor readings. There are several microcontrollers available. The Uno R3 has a universal serial bus (USB) interface to connect it to a computer, six analog input (A0–A5) and 14 digital input/output (I/O) pins (D0–D13) (Fig. 1A). The Mega 2560 R3 has additional analog (A0–A15) and digital pins [D0–D53 (Fig. 1B)] and can thus be used to build a system that uses more sensors and controls more valves. However, the basic operation of both microcontrollers is identical. The analog pins can be used to measure sensors with a voltage output (0–5 V DC). These microcontrollers accepts inputs from a variety of analog

Table 1. Equipment list and prices (Oct. 2014) for the components needed to build an irrigation system. The system uses a microcontroller to measure capacitance soil moisture sensors, and can then open or close irrigation valves as needed.

Equipment[z]	Brand	Price
	Hardware	
Uno R3 microcontroller	Arduino, Ivrea, Italy	$25
Secure digital (SD) shield 2.0	Adafruit, New York, NY	$20
Shield stacking headers for Arduino (R3 Compatible)	Adafruit	$2
1 Gigabyte SD card	SanDisk, Milpitas, CA	$10
8-module 5-V direct current (DC) relay driver	SainSmart, Leawood, KS	$15
10HS Soil moisture sensor	Decagon Devices, Pullman, WA	$110
6- to 18-V DC 1-inch latching solenoid valve	Galcon, San Rafael, CA	$30
24-V alternating current 1-inch solenoid valve	Rain Bird, Tucson, AZ	$25
5-mm light-emitting diode	Generic	$1
9-V DC battery	Generic	$1
	Wiring	
A/MicroB universal serial bus cable	Generic	$4
Male/Male jumper wires[y]	Generic	$8
Female/Male jumper wires[y]	Generic	$8
Female/Female jumper wires[y]	Generic	$8
3-mm "euro-style" terminal block connector	Generic	$5

[z] 1 inch = 2.54 cm, 1 mm = 0.0394 inch.

[y] 0.1 inch header × 6 inch length.

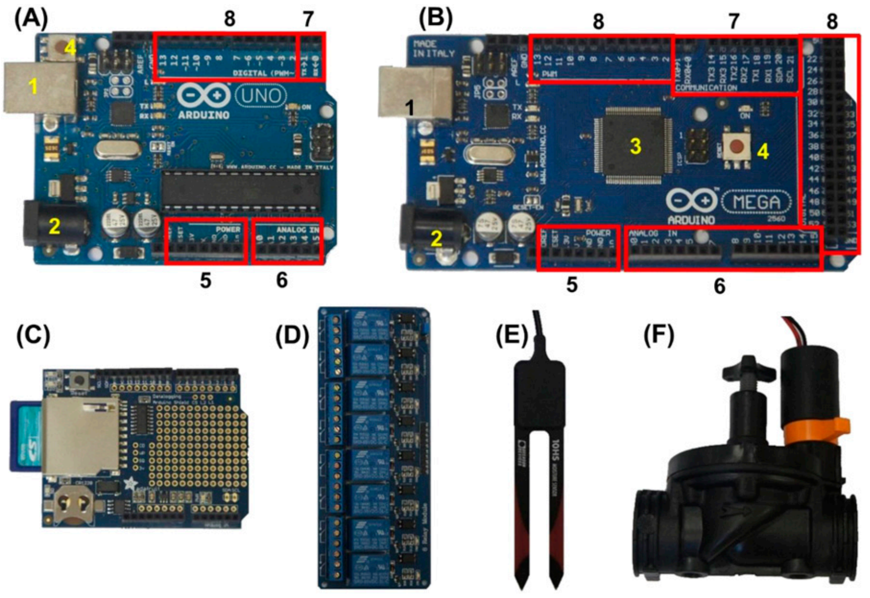

Fig. 1. Open-source prototype boards (A) Uno R3 and (B) Mega 2560 R3 (both from Arduino, Ivrea, Italy), where: 1) universal serial bus (USB) jack, 2) power jack, 3) processor, 4) reset button, 5) power pins, 6) analog inputs, 7) communication pins for data transmission (TX) and reception (RX), 8) digital inputs/outputs, (C) secure digital (SD) shield (SD 2.0; Adafruit, New York, NY), (D) 5-V direct current (DC) relay board with eight separate relays (SainSmart, Leawood, KS), (E) soil moisture sensor (10HS; Decagon Devices, Pullman, WA). (F) 6- to 18-V DC 1-inch (25.4 mm) latching solenoid valve (Galcon, San Rafael, CA).

(using the analog pins) and digital sensors [using communication pins for data transmission (TX) and reception (RX)], such as temperature, humidity, light, movement, level, soil moisture, and electrical conductivity. The digital pins can be used to provide 5 V DC to external equipment, such as actuators, relays, small motors, and LEDs. The maximum current for these pins is 40 mA and the digital pins can thus not be used to power large equipment. However, the pins can be used to control relays, which in turn can control equipment with much larger power requirements.

The microcontrollers can be powered directly from the USB port on a computer using a USB cable or from a DC power supply or battery (7–12 V DC). When using a battery, users need to keep in mind that the microcontroller uses relatively much power. Our testing with the set up described here indicates that an Uno R3 will drain a standard 9-V battery in ≈4 h. The use of rechargeable batteries powered by solar panels or sleep-mode libraries to reduce the power consumption are useful in situations where power supply is a concern. The microcontroller itself is not able to store data. For data collection, an SD shield (SD 2.0, Adafruit) with SD card can be connected (“stacked”) onto the microcontroller. The SD shield also has a RTC, keeping track of time, even when the microcontroller is not powered. Digital pin 10 is used for communication between the SD shield and the microcontroller.

The relay driver (SainSmart) had eight relays, which can be controlled using digital pins on the microcontroller. The relays are used to control power going to the solenoid valves.

Wiring

Wiring connections were made using jumper wires (connections to and from the microcontroller) or screw terminals (for connections to the relays). In a permanent system, we recommend either soldering the wires directly to the boards or using plug-in shields that convert the pins on the microcontroller to screw terminals [e.g., ScrewShield (ITEAD Studio, Shenzhen, China) or ProtoScrewShield (Sparkfun, Niwot, CO)]. This allows for more secure connections than the use of jumper wires.

The relay driver board ground (GND) and 5V terminals were connected to the GND and 5V terminals on the microcontroller to power the board. The relay driver board also has eight connectors that are used to control the eight relays (labeled K1–K8). The control connectors for relays 1 and 4 were connected to microcontroller digital pin D4, those for relays 2 and 3 to D5, relay 5 to D6, relay 6 to D7, and relay 7 to D8 (Fig. 2). Pins D4 and D5 each controlled two relays to switch power to a latching solenoid valve.

Solenoid valves were connected to the relay driver board. One latching solenoid valve was connected to relays 1 to 4 to create an H-Bridge (Fig. 3) with a 9-V DC battery. An H-Bridge can apply DC power to a circuit (in this case a latching solenoid valve) with regular and reversed polarity (in this case +9 and –9 V DC from the battery). Latching solenoid valves require only a short pulse of a positive voltage (50 ms) to open and a pulse with negative voltage to close. These valves have an internal magnet, which holds the solenoid in position when no power is applied. Latching valves are ideal for situations in which electrical power is limiting, and a 9-V DC battery can provide enough energy for several thousand valve cycles (data not shown).

The negative terminal of the 9-V DC battery was connected to the common (COM) terminal of relays 2 and 4, and the positive terminal of the battery was connected to the normally open (NO) terminal of relays 1 and 3. Jumper wires connected the NO terminal of relay 2 to the COM terminal of relay 1 and the NO terminal of relay 4 to the COM terminal of relay 3 (Fig. 3). To open the latching valve, we connected COM and NO terminals of relays 1 and 4 for 50 ms, which applied +9 V DC to the latching valve. To close the valve, –9 V DC was applied to the solenoid by connecting the COM and NO terminals of relays 2 and 3 for 50 ms.

We used relays 5, 6, and 7 to open and close each of the three 24-V AC solenoid valves, which require constant power to remain open. One wire from the 24-V AC power supply was split into three wires, which were connected to one of the wires from valves 2, 3, and 4. The other wire from the 24-V AC power supply also was divided into three wires, which were connected to the COM terminal of relays 5, 6, and 7. The other wire

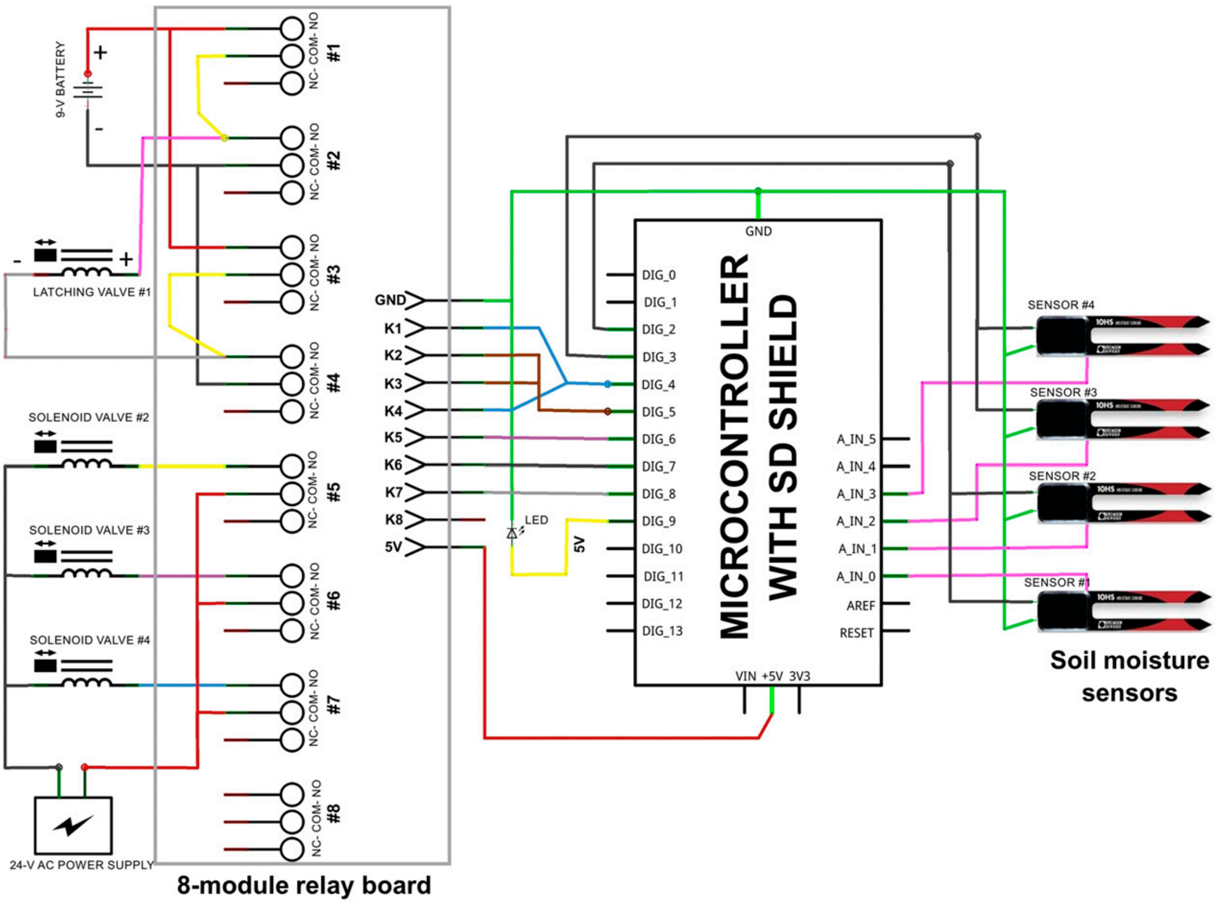

Fig. 2. Wiring diagram to connect four capacitance sensors and a 5-V direct current (DC) relay board with eight relays to a microcontroller. Four of the relays are used to control power going to a latching solenoid valve, while three relays are used to control three regular solenoid valves. Relay 8 is not used. K1–K8 indicate the eight pins on the relay board that control relay no. 1–no. 8. Applying 0 V DC to any of these pins will connect the common (COM) and normally open (NO) terminals of the associated relay, allowing current to flow. Applying 5 V DC disconnect the COM and NO terminals, stopping current flow.

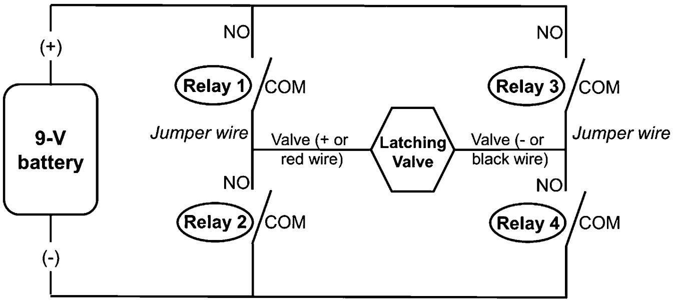

Fig. 3. Electrical diagram of an H-bridge with four relays, designed to control power to a latching solenoid valve. When the common (COM) and normally open (NO) terminals of relays 1 and 4 are connected (and those of relays 2 and 3 are not), +9 V direct current (DC) will be applied across the latching valve, opening the valve. When the COM and NO terminals of relays 2 and 3 are connected, −9 V DC is applied, closing the valve.

from valves 2, 3, and 4 was connected to the NO terminal in relays 5, 6, and 7 (Fig. 2). That allowed relays 5, 6, and 7 to control the power going to valves 2, 3, and 4.

The voltage output wires (red) from the four moisture sensors were connected to analog pins A0 to A3 of the SD shield, the excitation/power wires (white) were connected to digital pins D2 (sensors 1 and 2) and D3 (sensors 3 and 4), and all ground wires (bare) were connected to GND. An LED was connected to digital pin D9 and GND, with the shorter, negative lead of the LED to GND (Fig. 2).

Software

The program was written using the Arduino integrated development environment (IDE) freeware [version 1.0.5, Arduino (Arduino, 2014a)] using the C/C++ programming language. For more information about the software, visit the company website (Learning and Forum Menus, Arduino). The Arduino driver needs to be installed to be able to connect the microcontroller to a computer (Arduino, 2014b). The SD and RTClib libraries (equivalent to subroutines) need to be installed to enable use of the SD shield (Adafruit, 2014b). A tutorial for installing libraries is available from the manufacturer (Adafruit, 2014a). For communication between the computer and the microcontroller, the appropriate board and serial port need to be selected in the IDE (under "Tools").

The complete irrigation controller program is available online (University of Georgia, 2014) along with a pictorial guide on the assembly of the hardware. In the program, we read the output signal from the four soil moisture sensors, convert these values into a voltage, and then into VWC using a substrate-specific calibration equation for commercial peat:perlite medium (Fafard 1P; Fafard, Agawam, MA). The system controls irrigation by opening a valve when VWC readings drop below irrigation thresholds. Finally, measurements are saved on an SD card and shown on the computer screen (Serial monitor). To accomplish these tasks, the program was divided in three parts: 1) declaration of constants, variables, and libraries, 2) a "setup" routine, which runs once when the program first starts, and 3) a "loop" routine, which contains the main part of the program and runs indefinitely.

Libraries to perform common tasks are available from both the microcontroller and the SD shield manufacturers. We used the built-in SPI and WIRE libraries (both for communication between the microcontroller and SD shield), and the SD and RTClib libraries from the SD shield manufacturer (Adafruit). The SD library facilitates the use of the SD card, and the RTClib library is needed to use the RTC built into the SD shield and keeps track of time.

The setup routine allows users to specify the time interval at which the main part of the program is executed, how long valves are opened when irrigation is needed, irrigation thresholds for the different plots, and calibration constants for the soil moisture sensor. All of these variables can easily be changed by users. The next section of the setup routine typically does not need to be modified by users. It specifies communication parameters between the computer and the microcontroller, sends error messages to the computer screen in case of common programming or hardware problems, and configures some of the digital pins as output so that they can be used to provide 0 or 5 V DC to the SD shield, soil moisture sensors, relays, and LED as needed. Finally, the setup routine specifies that the internal-regulated 1.1-V DC voltage of the microcontroller should be used as the reference for all voltage measurements using the analog pins. When taking a voltage measurement, the microcontroller assigns a value from 0 to 1023 to the measured voltage. With a 1.1-V DC reference voltage, a raw value of 0 corresponds to 0 V DC and 1023 to 1.1 V DC. Knowing the reference voltage for these measurements allows for conversion of the raw value to the actual voltage signal from the sensors.

The loop is the main part of the program that measures the soil moisture sensors and controls the irrigation. To measure sensor 1 and 2, digital pin D2 is set high, which provides 5 V DC to power both sensors. The raw sensor output from the two sensors is measured after a 10-ms delay (according to manufacturer's guidelines) on analog pins A0 and A1, after which the power to the sensors is turned off. This is achieved with the instructions below. A similar set of instructions is then used to measure sensors 3 and 4.

```
digitalWrite(2, HIGH);
delay(10);
sensorValue[1] = analogRead(A0);
sensorValue[2] = analogRead(A1);
digitalWrite(2, LOW);
```

The analog sensor readings (0 to 1023) are then converted to a voltage output (0 to 1.1 V), which is used to calculate the VWC using a calibration equation. The sensor manufacturer provides a generic calibration equation for soilless substrates, but substrate-specific calibrations can improve the measurement accuracy (van Iersel et al., 2013). Guidelines for calibrations are provided by Cobos and Chambers (2010) and Kargas and Soulis (2012). We followed these guidelines to determine a multipoint calibration for the peat:perlite substrate used in this study, using three sensors. Since there were no statistically significant differences among the calibrations of the three sensors, one general calibration equation for 10HS sensors in peat:perlite was developed. The following instruction shows how the raw sensor value for the first sensor is converted to a voltage and then multiplied by the slope of the calibration line (SubCalSlope = 1.1785). Subsequently, the intercept of the calibration line (SubCalIntercept = –0.4938) is added to calculate the VWC. Similar instructions are used for the other sensors.

```
VWC[1] = sensorValue[1] * (1.1/1023.0)
         * SubCalSlope
         + SubCalIntercept
```

The program then checks if the measured VWCs are reasonable (0 < VWC < 0.8). If not, the program then briefly sets digital pin D9 high, which applies 5 V DC to the LED, turning it on. This provides a quick visual way to determine whether any of the sensors may not be working.

After all VWCs have been determined, the program determines whether the VWC in any of the four plots is below the specific threshold for that plot. If so, the associated solenoid valve is opened for the specified period of time. This is done by using the relays to control the power to the solenoids. Note that the relays use "reverse logic", meaning that setting a digital pin low and applying 0 V DC to the control pin for a relay will cause the COM and NO terminals of the relay to be connected, and allow power to be applied to the solenoid. Conversely, setting a digital pin high will apply 5 V DC to the control pin for the relay, disconnecting the COM and NO terminals of the relay.

Latching and regular solenoid valves need to be controlled differently. For the latching valve on plot 1, digital pin D4 controls both relays 1 and 4 on the relay driver board. Setting that pin low causes the COM and NO terminals in relays 1 and 4 to be connected, applying 9 V DC from the battery to the solenoid (Fig. 3). After a 50-ms delay, digital pin 4 is set high, disconnecting the COM and NO terminals, and removing the power from the solenoid. This 50-ms pulse of 9 V DC to the solenoid will cause the solenoid to open. The microcontroller then sends a message to the computer screen indicating that irrigation of this plot has been started. It also adds 1 to a counter (Counter[1]). That allows users to keep track of how often different plots are irrigated. The program then has a delay that causes the program to pause for a period that is equal to the specified irrigation time (IrrigTime). Note that IrrigTime needs to be multiplied by 1000 because users specify the desired irrigation time in seconds, but the delay instruction uses units of milliseconds. The following instructions are used to achieve this:

```
if (VWC[1] <= Threshold[1]) {
digitalWrite(4, LOW);
delay(50);
digitalWrite(4, HIGH);
Serial.print(“Plot 1 irrigation started.”);
Counter[1] = Counter[1] + 1;
delay(IrrigTime*1000.0);
```

After the desired irrigation time, the latching valve needs to be closed, which is done by applying 9 V DC with the opposite polarity to the solenoid. This uses digital pin D5, which is set low for 50 ms, connecting the COM and NO terminals on relays 2 and 3 and applying –9 V DC to the solenoid causing it to close (Fig. 3).

```
digitalWrite(5, LOW);
delay(50);
digitalWrite(5, HIGH);
}
```

Controlling regular solenoid valves is simpler. Connecting the COM and NO terminals on the associated relay will allow power to be applied to the solenoid, opening it. Disconnecting the COM and NO terminals of the relay will close the valve. The following instructions determine whether plot 2 needs to be irrigated, and if so, apply power to the relay (digital pin 6 is used to control this relay), write a message to the computer screen, count how often the plot has been irrigated, wait for the designated irrigation time, and then close the valve.

```
if (VWC[2] <= Threshold[2]) {
digitalWrite(6, LOW);
Serial.print(“Plot 2 irrigation started.”);
Counter[2] = Counter[2] + 1;
delay(IrrigTime*1000.0);
Serial.println(“Irrigation finished”);
digitalWrite(6, HIGH);
}
```

The final part of the program then writes the collected information to the SD card to store the data in comma-separated values (CSV) format. The SD card needs to be removed from the SD shield for data retrieval and the CSV format makes it easy to import the data into a spreadsheet for processing and graphing. In principle, the data can also be transferred to a computer in real time using Wi-Fi, radio, Bluetooth, or general packet radio service (GPRS, used for cellular communication). For example, Bitella et al. (2014) successfully used GPRS for wireless data transmission to a computer. We have successfully tested the use of radio to transmit VWC data from a microcontroller to a computer (R.S. Ferrarezi, R.M.L. Silva, S.K. Dove, and M.W. van Iersel, unpublished data). Real-time data transmission allows users to graph data continuously, for example, using freeware like Google charts (Google, 2014). However, continuous data transmission increases the cost, complexity, and power requirements of the system, so users need to decide whether these costs are outweighed by the advantage of easy real-time data access. After the data have been written to the SD card, the program pauses to assure that the loop part of the program runs at the user specified interval.

The program needs to be compiled after it has been written and if there are no errors, it can be uploaded to the microcontroller. This procedure has to be repeated every time changes are made in the program. Error messages will appear in orange in the left bottom corner of the IDE screen.

For simplicity, we have described an irrigation controller based on an Uno R3, controlling four individual plots. We also developed a larger system that monitors and irrigates 14 plots using 24-V AC solenoid valves and a larger microcontroller (Mega 2560 R3), which has 16 analog input pins and 54 digital I/O pins (Fig. 1B). The hardware, wiring diagram, program and step-by-step pictorial guide are also available online (University of Georgia, 2014).

Trial with potted *Hibiscus*

We conducted a trial using four ‘Panama Red’ hibiscus plants cultivated in a peat:perlite substrate (Fafard 1P) in 8-inch pots. We made no attempt to choose plants with similar sizes, since that was not important for testing purposes. On average, plants had a height of 11 inches and were 22 inches wide, with about eight branches each. Plants were pinched to increase branching 2 weeks before the trial. Environmental conditions in the greenhouse were monitored using a temperature and relative humidity sensor (HMP50; Vaisala, Woburn, MA) and a photosynthetic photon flux (*PPF*) sensor (QSO-Sun; Apogee Instruments, Logan, UT) connected to a datalogger (CR10X; Campbell Scientific,

Table 2. Irrigation thresholds for four pots with ‘Panama Red’ hibiscus. Whenever the measured volumetric water content (VWC) of the substrate in the pots dropped below the irrigation threshold, an automated irrigation controller applied 33 mL (1.1 fl oz) of water. Values of VWC are means from the time that the VWC first reached the irrigation threshold until the end of that particular period.

Pot	Period	Irrigation threshold	Measured VWC (mean ± SD)
		$(m^3 \cdot m^{-3})^z$	
Pot 1	Day 1–21	0.2	0.22 ± 0.02
	Day 21–32	0.4	0.41 ± 0.03
	Day 32–41	0.5	0.51 ± 0.02
Pot 2	Day 1–21	0.3	0.30 ± 0.01
	Day 21–32	0.5	0.49 ± 0.03
	Day 32–41	0.4	0.41 ± 0.01
Pot 3	Day 1–21	0.4	0.43 ± 0.03
	Day 21–32	0.3	0.32 ± 0.02
	Day 32–41	0.2	0.20 ± 0.01
Pot 4	Day 1–21	0.5	0.52 ± 0.02
	Day 21–32	0.2	0.21 ± 0.01
	Day 32–41	0.3	0.31 ± 0.03

[z] 1 m^3 = 35.3147 ft^3.

Logan, UT). Readings were taken every 20 min and daily averages were recorded. The vapor pressure deficit (VPD) was calculated from the saturated and actual air vapor pressure, using the air temperature and relative humidity data. Daily light integral (DLI) was calculated by integrating the *PPF* measurements throughout the day. Values (mean ± standard deviation) for temperatures were 20.5 ± 0.8 °C, for relative humidity 54.5% ± 15%, for VPD 1.1 ± 0.4 kPa, and for DLI 8.4 ± 4.1 $mol·m^{-2}·d^{-1}$.

An irrigation manifold was built using 1-inch polyvinyl chloride (PVC) pipe connected to the solenoid valves and ¾-inch polyethylene tubing with $2\text{-}L·h^{-1}$ drip emitters (PCJ; Netafim, Fresno, CA) connected to a dribble ring with four holes (Dramm, Manitowoc, WI) to allow for even distribution of the water across the substrate surface. Water pressure was maintained at 25 psi using a pressure regulator, and three additional $8\text{-}L·h^{-1}$ emitters were installed on each irrigation line to ensure proper closing of the valves after an irrigation event. One moisture sensor was inserted diagonally at a 45° angle into each pot. The trial was started 12 d later, providing time for the substrate to settle around the sensor. Every time the VWC dropped below the VWC threshold for a specific pot, the respectively solenoid valve was opened for 1 min, applying 33 mL per irrigation. Irrigation threshold was changed to 0.2, 0.3, 0.4, or 0.5 $m^3·m^{-3}$ on days 1, 21, and 32 (Table 2).

The microcontroller was connected to a laptop with a USB cable, which was used to power the controller and for communication between the microcontroller and laptop. This allowed us to monitor the data from the microcontroller in real time on the laptop.

Irrigation control

The system effectively monitored and controlled irrigation for most of 41-d trial. When the irrigation threshold was lower it took anywhere from less than 1 (pot 3, day 21) to 5 d (pot 1, day 1–5; pot 4, day 21–26) for enough evapotranspiration to lower the VWC to the threshold. During this

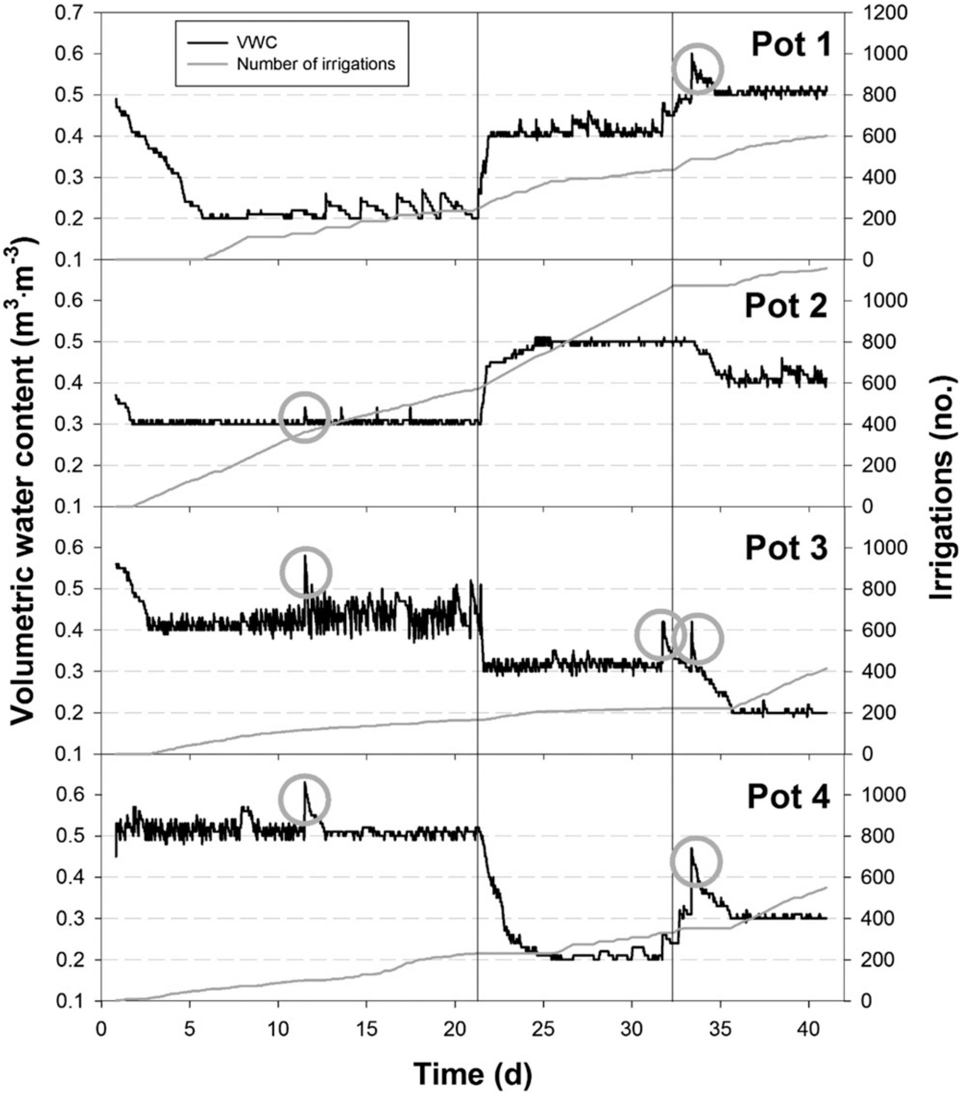

Fig. 4. Substrate volumetric water content (VWC) and the cumulative number of irrigations of pots with 'Panama Red' hibiscus in an automated irrigation system monitored with capacitance soil moisture sensors (10HS; Decagon Devices, Pullman, WA). Irrigation was triggered when substrate VWC dropped below set thresholds. Gray circles indicate times when hardware or software changes were made, resulting in unneeded irrigations. Irrigation thresholds were changed on days 1, 21, and 32, as indicated by the vertical lines. See Table 2 for irrigation thresholds.

period, the number of irrigations did not increase (Fig. 4). After the threshold was reached, the controller was able to maintain the VWC close to the threshold, typically within 0.01 to 0.04 $m^3 \cdot m^{-3}$ (Table 2, Fig. 4). The measured VWC in pot 3 initially was more variable than that in other pots, likely because one of the holes in the dribble ring was directly above the soil moisture sensor. That caused much of the applied water to quickly reach the sensor, causing a relatively large and rapid increase in measured VWC. Repositioning the dribble ring on day 21 reduced the variability in VWC (Table 2, Fig. 4).

When the irrigation threshold was increased, the number of irrigations increased quickly as well, as the controller irrigated often to increase the VWC to the threshold (Fig. 4). The number of irrigations differed greatly among the pots, likely because of differences in plant size, with larger plants using more water, and thus requiring more frequent irrigation. We made changes in the program and wiring that caused unneeded irrigation events at day 11.4 (pots 2, 3, and 4), 31.7 (pot 3) and 33.3 (pots 1, 3, and 4) (Fig. 4), resulting in temporary increases in VWC.

Similar results have been previously reported in studies using sensor-based irrigation with much more expensive dataloggers (Campbell Scientific) (Bayer et al., 2013; Nemali and van Iersel, 2006; van Iersel et al., 2010). However, in this study, we did not see that VWC was more variable at low VWC thresholds, contrary to previous studies (Nemali and van Iersel, 2006; van Iersel et al., 2010). This difference is unlikely to be related to the irrigation controller itself since the basic principle of operation is the same for this controller and those used in previous work. Instead, the lack of larger fluctuations in VWC at lower irrigation threshold may be the result of the relatively large container volume, small irrigation volume applied at each irrigation, and the water-holding capacity of the peat:perlite. Different substrates, such as coconut coir or pine bark, might affect the precision and variability of the VWC readings because differences in pore size distribution affect the hydraulic properties of the substrate. We recommend substrate-specific calibrations to improve the accuracy of the VWC readings.

Conclusions

The system effectively monitored and controlled container VWC during a 41-d trial, requiring little maintenance and irrigating the pots when the substrate VWC dropped below the set thresholds. This system maintained the VWC close to the thresholds throughout the study. This soil moisture sensor-based irrigation supplies water according to plant demand, which is useful for plant physiology and irrigation research as well as commercial applications. Open-source microcontrollers, such as the Uno R3 and Mega 2560 R3, are low cost, effective, and reliable. Thus, they can be used for precision irrigation control in many horticultural situations, from home gardens to automated production facilities.

Literature cited

Adafruit. 2014a. Arduino libraries. 28 Oct. 2014. <http://learn.adafruit.com/adafruit-all-about-arduino-libraries-install-use>.

Adafruit. 2014b. Downloads. 28 Oct. 2014. <https://learn.adafruit.com/adafruit-data-logger-shield/downloads>.

Allen, R.G., L.S. Pereira, D. Raes, and M. Smith. 1998. Crop evapotranspiration: Guidelines for computing crop water requirements. FAO Irr. Drainage Paper Bul. 56.

Arduino. 2014a. Download the Arduino software. 28 Oct. 2014. <http://arduino.cc/en/Main/Software>.

Arduino. 2014b. Getting started with Arduino. 28 Oct. 2014. <http://arduino.cc/en/Guide/HomePage>.

Bayer, A., I. Mahbub, M. Chappell, J. Ruter, and M.W. van Iersel. 2013. Water use and growth of *Hibiscus acetosella* 'Panama Red' grown with a soil moisture sensor-controlled irrigation system. HortScience 48:980–987.

Bitella, G., R. Rossi, R. Bochicchio, M. Perniola, and M. Amato. 2014. A novel low-cost open-hardware platform for monitoring soil water content and multiple soil-air-vegetation parameters. Sensors 14:19639–19659.

Blonquist, J.M., S.B. Jones, and D.A. Robinson. 2005. A time-domain transmission sensor with TDR performance characteristics. J. Hydrol. 314:235–245.

Burnett, S.E. and M.W. van Iersel. 2008. Morphology and irrigation efficiency of *Gaura lindheimeri* grown with capacitance sensor-controlled irrigation. HortScience 43:1555–1560.

Cayanan, D.F., M. Dixon, and Y. Zheng. 2008. Development of an automated irrigation system using wireless technology and root zone environment sensors. Acta Hort. 797:167–172.

Chappell, M., S.K. Dove, M.W. van Iersel, P.A. Thomas, and J. Ruter. 2013. Implementation of wireless sensor networks for irrigation control in three container nurseries. HortTechnology 23:747–753.

Cobos, D.R. and C. Chambers. 2010. Calibrating ECH_2O soil moisture sensors. 26 Aug. 2014. <http://www.decagon.com/assets/Uploads/13393-04-Calibrating ECH2OSoilMoistureProbes.pdf>.

Conover, C.A. and R.T. Poole. 1992. Effect of fertilizer and irrigation on leachate levels of NH4-N, NO3-N, and P in container production of *Nephrolepis exaltata* 'Fluffy Ruufle' J. Environ. Hort. 10:238–241.

Crespo, J.M. and M.W. van Iersel. 2011. A calibrated time domain transmissometry soil moisture sensor can be used for precise automated irrigation of container-grown plants. HortScience 46:889–894.

Ferrarezi, R.S., M.W. van Iersel, and R. Testezlaf. 2014. Subirrigation automated by capacitance sensors for salvia production. Horticultura Brasileira 32:314–320.

Google. 2014. Google charts. 30 Oct. 2014. <https://developers.google.com/chart>.

Jones, H.G. 2004. Irrigation scheduling: Advantages and pitfalls of plant-based methods. J. Expt. Bot. 55:2427–2436.

Kargas, G. and K.X. Soulis. 2012. Performance analysis and calibration of a new low cost capacitance soil moisture sensor. J. Irr. Drain. Eng. 138:632–641.

Lea-Cox, J.D., W.L. Bauerle, M.W. van Iersel, G.F. Kantor, T.L. Bauerle, E. Lichtenberg, D.M. King, and L. Crawford. 2013. Advancing wireless sensor networks for irrigation management of ornamental crops: An overview. HortTechnology 23:717–724.

Majsztrik, J., A.G. Ristvey, and J.D. Lea-Cox. 2011. Water and nutrient management in the production of container-grown ornamentals. Hort. Rev. 38:253–296.

Nemali, K.S., F. Montesano, S.K. Dove, and M.W. van Iersel. 2007. Calibration and performance of moisture sensors in soilless substrates: ECHO and Theta probes. Sci. Hort. 112:227–234.

Nemali, K.S. and M.W. van Iersel. 2006. An automated system for controlling drought stress and irrigation in potted plants. Sci. Hort. 110:292–297.

Shock, C.C. and F.-X. Wang. 2011. Soil water tension, a powerful measurement for productivity and stewardship. HortScience 46:178–185.

Silva, D., G. Oliveira, R. Silva, C. Fernandes, L. de Jesus, and I. Bergier. 2013. Automated control of soil moisture with solar energy for small rural producers. 6° Simpósio sobre recursos naturais e socioeconômicos do Pantanal. 1 Sept. 2014. <http://ainfo.cnptia.embrapa.br/digital/bitstream/item/93857/1/RE06.pdf>.

University of Georgia. 2014. Building irrigation controllers. 28 Oct. 2014. <http://hortphys.uga.edu/irrigationcontrol.html>.

U.S. Department of Agriculture. 2010. Census of horticultural specialties (2009). Volume 3, Special studies, Part 3. 1 Sept. 2014. <http://www.agcensus.usda.gov/Publications/2007/Online_Highlights/Census_of_Horticulture_Specialties/HORTIC.pdf>.

van Iersel, M.W., M. Chappell, and J.D. Lea-Cox. 2013. Sensors for improved efficiency of irrigation in greenhouse and nursery production. HortTechnology 23:735–746.

van Iersel, M.W., S. Dove, J.G. Kang, and S.E. Burnett. 2010. Growth and water use of petunia as affected by substrate water content and daily light integral. HortScience 45:277–282.